Information-Oriented and Energy-Aware Path Planning for Small Unmanned Aerial Vehicles

José Bento(✉), Meysam Basiri, and Rodrigo Ventura

Instituto Superior Técnico, Av. Rovisco Pais 1, 1049-001 Lisbon, Portugal
{jose.miguel.bento,meysam.basiri}@tecnico.ulisboa.pt,
rodrigo.ventura@isr.tecnico.ulisboa.pt

Abstract. Small Unmanned Aerial Vehicles (UAVs) are revolutionizing area coverage operations such as Search and Rescue (SAR) or surveillance and monitoring. However, maximizing the search efficiency in large areas with limited battery life and uncertain target locations remains a challenge. This paper aims to maximize the likelihood of finding the target as quickly as possible by introducing a novel formulation of the Coverage Path Planning problem that takes into account the energy constraints of a small UAV and prior information about the target area, employing a global optimization algorithm, specifically Simulated Annealing, to generate a path. The proposed algorithm can be applied to scenarios such as SAR, where the Area of Interest (AOI) cannot be fully covered by the UAV in a single flight due to its limited endurance, and where the expected target position is modeled by a Probability of Containment (POC) map. The algorithm prioritizes visiting the areas where the POC is high in order to increase the probability of detecting a target and reduce the time at which it is detected, which in SAR represents a higher likelihood of survival. The algorithm presented demonstrates superior performance compared to a baseline Boustrophedon algorithm typically used for area coverage.

Keywords: UAV path planning · Simulated Annealing · Search and Rescue · Energy-aware · Information-oriented

1 Introduction

Recent advancements in the design and production of small Unmanned Aerial Vehicles (UAVs) have made these platforms suitable for a wide range of applications, from Search and Rescue (SAR) operations in a maritime or terrestrial environment, to surveillance and monitoring missions. The use of UAVs reduces both the operational cost, as well as the inherent risk to human assets traditionally involved in these missions by eliminating the necessity for larger manned airborne platforms to assist in the search for a target. Small, low-flying UAVs can also be more time-efficient and effective than traditional means at finding targets on the ground in situations where the visibility from a high altitude is reduced due to poor atmospheric conditions such as fog or smoke.

M. F. Santos et al. (Eds.): EPIA 2024, LNAI 14969, pp. 78–89, 2025.
https://doi.org/10.1007/978-3-031-73503-5_7

The main contribution of this paper is a novel path planning algorithm for a SAR scenario that maximizes the likelihood of finding the target as quickly as possible by leveraging prior information about the estimated target position modeled by a POC map (information-oriented), while taking into account mission and system constraints such as restricted areas of operation and being energy-aware regarding flight-time limitations of small UAVs. This paper also presents illustrative examples that show the performance of our method when compared to other state-of-the-art algorithms.

We have performed a decomposition of the Area of Interest (AOI) into a regular grid made up of small cells, and to each one a value is attributed that translates the probability that the target is contained inside it. Subsequently, a discrete path planning algorithm is applied to find the optimal path given an objective function that translates the SAR mission goals.

Algorithms that guarantee the global optimality of the solution exist, such as A* with an admissible heuristic. However, despite this algorithm performing well for path planning problems where the objective is to go from point A to point B and the euclidean distance to the target can be used as an heuristic, in the problem described in this work, since the target position is unknown, finding an admissible heuristic is not as straightforward and calculating it is much more computationally expensive. This means that these approaches do not scale well for large AOIs and long paths. Algorithms such as Particle Swarm Optimization (PSO) have been used for path planning [12], but these are more suitable for continuous optimization. Therefore, in this paper, Simulated Annealing is used, since it demonstrates good scalability performance and for its ease of adaptation to a multi-UAV scenario that will be the future focus of this research.

This work is part of a larger project called Aero.Next [15] that is focused on the use of an air-deployed multi-UAV system for a wide range of operations. However, the work presented aims to optimize the path planning of a single small UAV equipped with a camera for target detection.

The paper is structured as follows: in Sect. 2 a literature review is conducted on methods for the decomposition of the AOI, strategies to leverage target information, energy models of small UAVs and a brief overview of Simulated Annealing is presented. In Sect. 3 the problem is formulated and an objective function is presented. In Sect. 4 a path planning algorithm is presented, the application of which is covered in Sect. 5. There, some tests are conducted to evaluate the effect of several algorithm parameters and these are applied to a complex mission scenario. Finally, Sect. 6 draws some conclusions and outlines possible future work.

2 Related Works

The problem of how to cover a given area while avoiding obstacles and minimizing completion time is the object of study of Coverage Path Planning (CPP). A classification for CPP problems was proposed in [4], where they are characterized by their coverage guarantees, available environment information, area decomposition method and number of vehicles involved in the coverage. If the position of obstacles is unknown at the start of the mission, which is usually the case for ground vehicles, an online CPP approach is needed. However, for UAVs flying at sufficient height, obstacle avoidance is not a big concern since No-Fly Zones (NFZs) can easily be defined by the operator, enabling the use of offline algorithms. Several authors also focus on the use of multiple vehicles to perform the coverage tasks, and in [8] the authors present an area division algorithm that considers the different energy capacities of the available UAVs.

The Area of Interest (AOI) is defined as the area that is intended to be covered by the search vehicles, whereas NFZs can be defined to represent areas where flight operations are restricted. The authors in [3] explore the different ways in which an AOI can be decomposed in order to allow for the application of path planning algorithms. Three methods can be employed: Exact cellular decomposition, Approximate cellular decomposition and No decomposition methods.

Exact cellular decomposition divides the area into smaller trapezoid-shaped sub-areas (cells) that are explored by the search vehicle in simple back-and-forth motions. The CPP problem is reduced to a motion planning problem between the cells, where the vehicle can only traverse between adjacent cells, which is the method applied in [2]. Due to the decomposition in large cells, this method does not allow for a detailed representation of a POC model of the target distribution.

Approximate cellular decomposition divides the AOI into small regular-sized cells, i.e. discretizing the AOI by applying a grid over it, resulting in an approximation of the target area. This method is used in [1], where the authors also optimize the grid placement by shifting and rotating it in order to maximize the area covered by the cells. A cell is considered to have been covered once the UAV enters it, since the size of the grid is selected to match the footprint of the detection system of the UAV. The generated path in this case consists of a set of waypoints that correspond to the position of the center of the cells.

No decomposition methods apply simple movements such as back-and forth or spiral patters to cover an area, but are not able to cope with complex-shaped areas or NFZs inside the AOI. This type of method is usually used in studies dedicated to optimizing the energy performance, such as the one in [6].

The authors in [12] focus on the surveillance problem of detecting and tracking moving ground targets inside a large AOI. The path planning algorithm prioritizes visiting areas where it is expected that the UAVs will have the most success at finding targets. The authors take into account prior information regarding the expected density of target distribution across the AOI and apply a PSO algorithm to generate the paths. In [14], the authors focus on a terrestrial SAR scenario where the objective is to locate missing people during a forest fire. They calculate a risk map that translates the danger posed by the fire and implement an Attraction algorithm (Att) to optimize the path such that zones with high risk are visited first in order to detect any targets that might be in immediate danger. Finally, the authors in [13] focus on the maritime SAR scenario, where the objective is to locate a target in distress drifting on the surface. They generate a Probability of Containment (POC) map that translates the probability that a certain cell contains the target and apply Dynamic Programming (DP) algorithms to generate the optimal path.

The biggest downside of using small electric UAVs is their limited flight time when compared to larger combustion engine platforms. The authors in [6] study the energy consumption of a quadcopter during several phases of flight and propose an energy-aware path planning method, which is then extended by the authors in [10] to multi-UAV path planning. The authors in [11] calculate an energy model for a quadcopter moving in a regular grid, making it suitable for use in an AOI decomposed with approximate cellular decomposition. The authors study the energy consumption for a straight-line segment at different speeds, as well as the energy necessary to perform turns.

Simulated Annealing (SA) draws inspiration from the process that occurs in the cooling of materials from liquid to solid state, in which randomly distributed particles gradually organize themselves in a well-defined structure [9]. Similarly to this process, SA starts with an initial guess of the solution and at each iteration it generates a neighbour to the current accepted solution. If this neighbour performs better than the accepted solution, it replaces it, but if it performs worse it is accepted with a probability that is directly proportional to the temperature. In other words, the higher the temperature, the more likely a worse solution is to be accepted and vice-versa. At the end of each iteration, the temperature is decreased by a cooling factor. After the temperature drops below a certain threshold, the algorithm stops and outputs the current accepted solution. The authors in [5] propose methods to calculate some SA parameters and define a termination criteria. An efficient implementation of SA is explored in [7], where a parallelization method is proposed.

3 Problem Formulation

The objective of this work is to develop a path planning algorithm that takes into account prior target information and energy limitations of the search platform, aiming to find a target as fast as possible in a SAR scenario. In this section, the problem is formalized in terms of its constraints regarding areas of operation and energy limitations, information available regarding estimated target distribution, and an objective function is formulated that translates the effectiveness of a path.

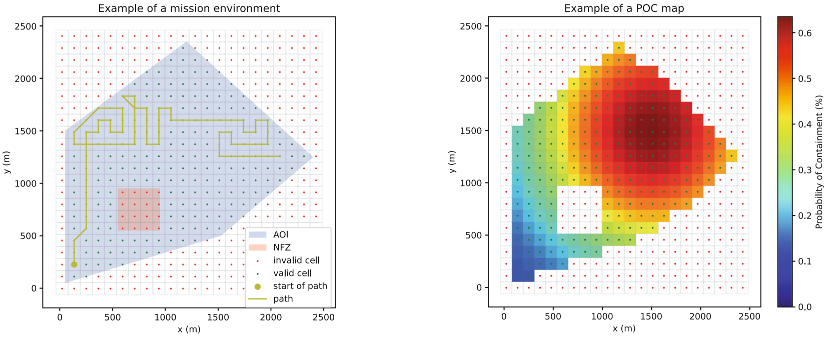
The Area of Interest (AOI) is defined by the operator as a polygon represented by a set of vertices in a two-dimensional plane and corresponds to the area that is intended to be covered by the UAV during the search. No-fly Zones (NFZs) are also defined by the operator as polygons represented by sets of vertices and correspond to areas where the UAV is not allowed to operate.

In this work, approximate cellular decomposition is applied to the AOI, since it provides flexibility for the path optimization, allows for a more detailed representation of the Probability of Containment (POC) map and facilitates the process of keeping track of which areas have been covered throughout the search. Therefore, it is necessary to define the size (d) of the square cells in the grid that will be used to decompose the AOI. It must allow for the UAV to fully cover a cell with its sensor footprint and detect any target present inside it. The operator should take into account the resolution of the camera, the size of the target and the atmospheric conditions when defining the height (h) at which the UAV should cover the terrain, since if the altitude chosen is too high, the camera might not be able to detect the target. The grid size d can be calculated using the formula:

$$d = 2(1 - p_o)h \tan\left(\frac{FOV}{2}\right) \tag{1}$$

where p_o is the percentage of overlap of the images taken by the camera (the percentage of the area of a cell that will be observed by covering one cell in its 4-connected neighbourhood) and FOV is the angular field of view of the camera. The overlap can be increased by the operator to increase the chances of target detection if the target is expected to be moving, since it is more likely to be detected by a UAV covering a neighbouring cell.

In Fig. 1a an example of an area operations is provided with a grid size that was calculated for $FOV = 84\,\text{deg}$, $h = 50$ m and $p_o = 50\%$. A cell is considered to be traversable by a UAV if it belongs to the set of valid cells $\mathcal{S}$, which is the case if its center is located inside the AOI and the cell is completely outside any NFZs. We consider that a UAV is only able to move between the centers of valid adjacent cells (in the 8-connected neighbourhood) and that the path starts at a user-defined position. The path generated by the algorithm is a sequence of the coordinates of the cell centers visited by the UAV. An example of a path is also provided in Fig. 1a.



(a) Example of an area of operations - the area in blue is the AOI and in red is the NFZ, the path is shown as a yellow line and the starting position as a yellow dot.

(b) Example of the prior information available in a SAR scenario - the color of each cell translates the probability that it contains the target.

Fig. 1. Example of a simple SAR mission scenario - the color of the center of the cells translates their status (green if it is valid and red otherwise) (Color figure online)

Prior information regarding the possible target position must also be considered, which for SAR scenarios takes the form a POC map. Each cell in $\mathcal{S}$

therefore has an associated probability that describes how likely the target is to be contained in it. The sum of the POC values of all the cells in $\mathcal{S}$ is equal to 1. An example of a POC map is presented in Fig. 1b. Finally, the energy limitations of the UAV are considered using the energy model presented in [11]:

$$E = 0.1164\ l + 0.0173\ \theta \tag{2}$$

where E is the energy consumption in kJ, l is the total length of the path in meters and θ is the sum of all the turn angles in degrees. The operator defines the maximum energy capacity of the UAV and the algorithm must ensure that the energy demands of the generated path do not exceed this value.

An objective function is defined in a way that translates the SAR mission objectives: finding the target, and doing so as fast as possible. The probability of detecting a target (D) given a certain path can be calculated by:

$$D = \sum_{i=1}^{k} P_i V_i \tag{3}$$

where k is the number of steps in the path, P_i is the POC value of the cell visited in the i^{th} step of the path and V_i is a boolean that is 0 if the cell visited in the i^{th} step has been visited before and 1 otherwise. The step at which a target is expected to be found is given by the average detection step (ADS):

$$ADS = \sum_{i=1}^{k} i P_i V_i \tag{4}$$

The objective function (J) used for the path optimization is given by:

$$J = \sum_{i=1}^{k} e^{-\epsilon i} P_i V_i \tag{5}$$

Analysing this function, it is clear that its value will be greater if more cells are visited (k is larger), if their POC is higher (P_i is larger), and if the number of revisited cells is lower ($V_i = 1$ more often), all of which are associated with a higher likelihood of finding the target. In order to reward paths that find the target sooner, the term $e^{-\epsilon i}$ was introduced, which decreases the contribution of cells visited later in the path. The decay factor ϵ defines how fast this decay occurs. The overall goal of the algorithm is thus to maximize J, indirectly maximizing D and minimizing ADS.

4 Proposed Method

Simulated Annealing was chosen as the optimization method for the problem formulated in this work for having good scalability performance, good performance in escaping local optima, and being easily adaptable for a scenario containing multiple UAVs that will be the subject of future work.

The pseudocode for the multi-threaded Simulated Annealing algorithm implemented in this work is presented in Algorithm 1. The inputs are a choice of algorithm $\mathcal{G}$ for the generation of the initial path guess, an initial temperature T_{init}, a cooling factor α, a minimum temperature T_{min}, the length of the Markov chains L_k, the number of threads $n_{threads}$ and a boolean $Sync$ that defines whether the threads synchronize their accepted paths between iterations of the Markov chains. The output is a path that consists of a list of cells to visit.

The initial guess can either be generated using the Attraction (Att) algorithm (an algorithm adapted from [14]) or the Random Walk (RW) algorithm. If the latter is used, a different initial guess is generated for each of the threads in order to increase diversity in the solutions generated by the individual threads. The function RUN_SA_IN_THREAD is responsible for running a single-threaded SA algorithm such as the one presented in [5]. If the user decides to use thread synchronization, after every thread finishes L_k iterations (reach the end of a Markov chain), they exchange solutions between them and start the next Markov chain with the best *AcceptedPath* found, as suggested in [7].

The function GENERATE_NEIGHBOUR is responsible for generating a neighbour path to the one provided as an argument. A neighbour path can be obtained by removing a step, changing the cell that is covered by a step, or adding another step in the middle of the path. After one of these operations has been carried out, if there is enough energy left-over for further steps at the end of the path, these are added.

5 Experiments and Results

5.1 Parameter Choice for Simulated Annealing

Several experiments were conducted to identify the combination of parameters for the SA algorithm that would best suit the use case covered here. All the tests presented in Sects. 5.1 and 5.2 are the result of running the Simulated Annealing algorithm for the scenario presented in Fig. 1 with 1000 kJ of available energy. For every parameter test, the algorithm is run 10 times and the average objective function value is shown in a solid blue line, whereas the deviations from this average are shown as a light-blue area. The average processing time required to compute the solutions is also shown in red.[1]

Tests for the initial temperature T_{init} were conducted and are displayed in Fig. 2 for the path initialization with the Attraction algorithm (Fig. 2a) and with Random Walk (Fig. 2b). The optimal values for T_{init} were found to be 0.0004 when using Att as an initial guess, and 0.0085 when using RW. Tests for the cooling factor α are presented in Fig. 3a. The processing time increases exponentially with the increase in the cooling factor, so a value of 0.96 was chosen, allowing for good results in an acceptable time-frame. The effect of the number of threads $n_{threads}$ was considered, and the results can be viewed in

[1] Processor: Intel(R) Xeon(R) Gold 6330 CPU @ 2.00GHz, Number of cores: 24, RAM: 215 Gb.

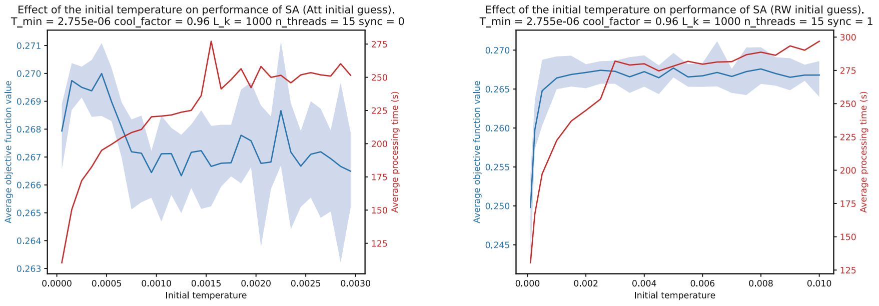
Algorithm 1. Multi-threaded Simulated Annealing algorithm

```
Inputs: $\mathcal{G}$, $n_{threads}$, $T_{init}$, $T_{min}$, $\alpha$, $L_k$, $Sync$

$p_{init} \leftarrow$ path generated with the algorithm specified by $\mathcal{G}$
for $idx = 0 : n_{threads}$ do
    RUN_SA_IN_THREAD($p_{init}$, $T_{init}$, $T_{min}$, $\alpha$, $L_k$, $Sync$)
end for
Wait for all threads to conclude
$path \leftarrow$ the path that performs best out of all threads
return $path$

function RUN_SA_IN_THREAD($p_{init}$, $T_{init}$, $T_{min}$, $\alpha$, $L_k$, $Sync$)
    $AcceptedPath \leftarrow p_{init}$
    $E_{accepted} \leftarrow -J(AcceptedPath)$    ▷ The goal is to maximize the objective function $J$ (5) and SA minimizes the solution energy $E$
    $T \leftarrow T_{init}$
    while $T > T_{min}$ do
        for $l = 0 : L_k$ do
            $CandidatePath \leftarrow$ GENERATE_NEIGHBOUR($AcceptedPath$)
            $E_{candidate} \leftarrow -J(CandidatePath)$
            if random([0,1]) ¡ exp $\left(\frac{E_{accepted} - E_{candidate}}{T}\right)$ then
                $AcceptedPath \leftarrow CandidatePath$
                $E_{accepted} \leftarrow E_{candidate}$
            end if
        end for
        if $Sync$ then
            Wait for all threads to conclude the current iteration
            $AcceptedPath \leftarrow$ the accepted path with the lowest energy of all threads
        end if
        $T \leftarrow \alpha T$
    end while
    return $AcceptedPath$
end function
```

Fig. 3b. It was found that there is no significant benefit to using more than 15 threads. Tests indicate that increasing L_k generates better results at the cost of more computational time and a value of 1000 was used. The tests for T_{min} found that it did not impact the performance significantly and a value of 2.755×10^{-6} was chosen. Finally, synchronization tests revealed it is most advantageous when the initial guess is generated using RW rather than with Att.

5.2 Effect of the Decay Factor in the Objective Function

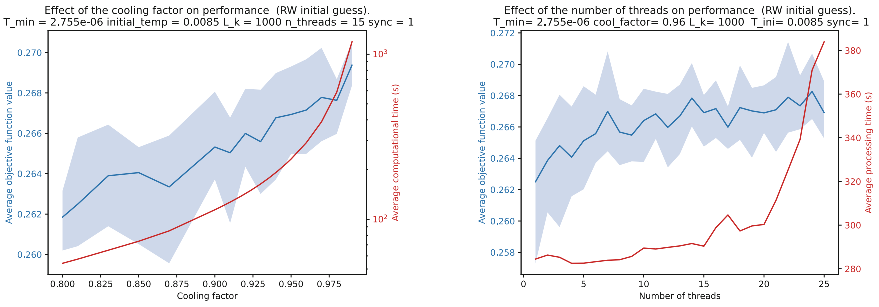
The objective function, defined in Sect. 3, represents both of the sub-objectives of detecting the target and doing so as fast as possible. In order to establish a trade-off between these two objectives, a decay factor ϵ was included in the objective function J (5). The lower this value, the less important is finding a



(a) Initial path guess using Att

(b) Initial path guess using RW

Fig. 2. Effect of the initial temperature T_{init} on the performance of SA. The average over 10 tests of the objective function (left vertical axis) is plotted in a blue line and the computational time (right vertical axis) is shown in a red line. (Color figure online)



(a) Effect of the cooling factor

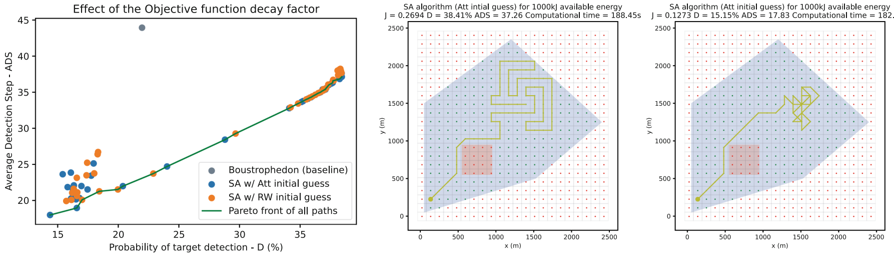
(b) Effect of the number of threads

Fig. 3. Effect of the cooling factor α and number of threads $n_{threads}$ in the performance of SA. The average over 10 tests of the objective function is plotted in a blue line and the computational time is shown in a red line. (Color figure online)

target fast, given by the average detection step ADS (4), which allows the SA algorithm to optimize the path in order to improve the target detection D (3).

In order to study the impact of the decay factor, tests were conducted where the SA algorithm was run for different values of ϵ for the scenario presented in Fig. 1 with 1000 kJ of energy available and the results are presented in Fig. 4a. As expected, paths calculated using lower values of ϵ result in higher D and higher ADS, while paths calculated using higher ϵ resulted in lower D and lower ADS. A Pareto front is also calculated for all the results and since it is only comprised of paths generated with the SA algorithm, it demonstrates that this algorithm produces better results than the baseline Boustrophedon algorithm.

It was determined that a decay factor of 0.01 results in an acceptable trade-off between the two sub-objectives and this is the value used throughout the tests in Sects. 5.1 and 5.3. The path generated using SA for this decay factor can be seen in Fig. 4b. In contrast, the result of using a decay factor of 0.45 can be observed in Fig. 4c, where the algorithm generates a path that focuses on minimizing the ADS at the cost of a lower D. This parameter should therefore be chosen by the operator to suit the operational needs.

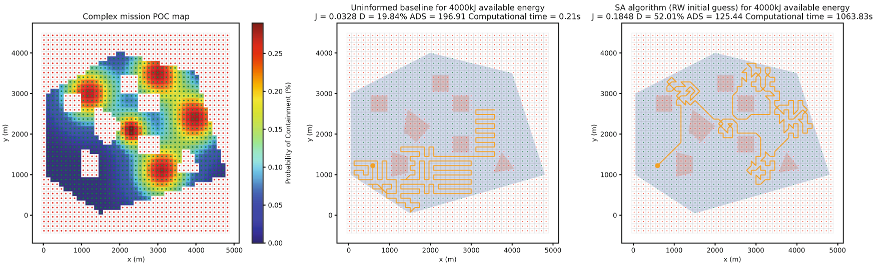


(a) Path performance for several ϵ (b) Path for $\epsilon = 0.01$ (c) Path for $\epsilon = 0.45$

Fig. 4. Effect of the decay factor ϵ on the SA algorithm performance regarding the trade-off of the sub-objectives defined in Eqs. (3) and (4)

5.3 Implementation Results for a Complex Mission Scenario

In order to test the scalability of the algorithms presented, a more complex mission scenario was tested. In Fig. 5a the POC map for this scenario is presented and in Fig. 5b the AOI and NFZs are shown, along with the result of applying a baseline Boustrophedon algorithm. The result of applying the Simulated Annealing algorithm is presented in Fig. 5c, where the initial guess was generated using RW with no thread synchronization and with the parameters discussed in Sect. 5.1.


Complex mission POC map
Probability of Containment (%)
0.25
0.20
0.15
0.10
0.05
0.00
Uninformed baseline for 4000kJ available energy
J = 0.0328 D = 19.84% ADS = 196.91 Computational time = 0.21s
SA algorithm (RW initial guess) for 4000kJ available energy
J = 0.1848 D = 52.01% ADS = 125.44 Computational time = 1063.83s
y (m)
x (m)
4000
3000
2000
1000
0
0
1000
2000
3000
4000
5000

(a) POC map (b) Baseline algorithm (c) Simulated Annealing

Fig. 5. Complex mission scenario with a large AOI and several NFZs

Table 1. Algorithm performance for a complex mission scenario ($\epsilon = 0.01$)

Algorithm	J	D (%)	ADS	Computational time[1] (s)
Boustrophedon (baseline)	0.0328	19.84	196.91	**0.21**
Attraction [14]	0.1389	41.27	142.01	1.66
SA with Att initial guess	0.1693	49.98	132.97	824.77
SA with RW initial guess	**0.1848**	**52.01**	**125.44**	1063.83

A comparison of the results of applying the algorithms discussed to this complex mission scenario are presented in Table 1. The SA algorithm with RW initial guess performed better than the baseline and the Attraction algorithms, detecting more targets (higher D), detecting them sooner on average (lower ADS), and generating a path with a higher objective function value J, which comes at the cost of more computational time.

6 Conclusions

In this work, a path planning algorithm is presented with the goal of maximizing the likelihood of finding a target as fast as possible in a SAR scenario. It takes into account the energy limitations of the small electric UAVs used for the search, and prior information regarding the search area in the form of a target POC map.

Firstly, a decomposition method for the AOI was implemented that takes into account the mission requirements and available UAV systems. Secondly, an objective function was constructed to represent the goals of a SAR mission regarding target detection and the time at which it takes place, with a parameter that allows the operator to select the desired trade-off between them. Subsequently, a Simulated Annealing algorithm was used to perform the path optimization and several tests were conducted in order to optimize the SA parameters for this specific use case. Finally, the algorithm was tested for a complex mission scenario, outperforming the Boustrophedon path usually used in coverage missions.

This paper leaves some areas open for study, such as the optimization of the deployment position and the extension of the algorithm for multiple UAVs. An implementation of this algorithm into an existing ground-station software can also be explored in order to facilitate its integration into existing systems.

Acknowledgement. This work was supported by the Aero.Next project (PRR - C645727867- 00000066) and LARSyS funding (DOI: 10.54499/LA/P/0083/2020, 10.54499/UIDP/50009/2020 and 10.54499/UIDB/50009/2020).

Disclosure of Interests.. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Apostolidis, S., Kapoutsis, P., Kapoutsis, A., Kosmatopoulos, E.: Cooperative multi-UAV coverage mission planning platform for remote sensing applications. Auton. Robots **46**, 1–28 (2022). https://doi.org/10.1007/s10514-021-10028-3
2. Bähnemann, R., Lawrance, N., Chung, J.J., Pantic, M., Siegwart, R., Nieto, J.: Revisiting boustrophedon coverage path planning as a generalized traveling salesman problem. In: Ishigami, G., Yoshida, K. (eds.) Field Serv. Robot., pp. 277–290. Springer, Singapore (2021). https://doi.org/10.1007/978-981-15-9460-1_20
3. Cabreira, T., Brisolara, L., Ferreira Jr, P.: Survey on coverage path planning with unmanned aerial vehicles. Drones **3**, 4 (2019). https://doi.org/10.3390/drones3010004
4. Choset, H.: Coverage for robotics - a survey of recent results. Ann. Math. Artif. Intell. **31**, 113–126 (2001). https://doi.org/10.1023/A:1016639210559
5. Delahaye, D., Chaimatanan, S., Mongeau, M.: Simulated annealing: from basics to applications. Handbook of Metaheuristics, pp. 1–35 (2019). https://doi.org/10.1007/978-3-319-91086-4_1
6. Di Franco, C., Buttazzo, G.: Energy-aware coverage path planning of UAVs. In: Proceedings - 2015 IEEE International Conference on Autonomous Robot Systems and Competitions, ICARSC 2015, April 2015. https://doi.org/10.1109/ICARSC.2015.17
7. García-Rodríguez, J., Vázquez, C., López-Salas, J.G., Ferreiro, A.: An efficient implementation of parallel simulated annealing algorithm on GPUs. J. Global Optim. **57**, 863–890 (2012). https://doi.org/10.1007/s10898-012-9979-z
8. Kapoutsis, A., Chatzichristofis, S., Kosmatopoulos, E.: DARP: divide areas algorithm for optimal multi-robot coverage path planning. J. Intell. Robot. Syst. **86**, 663–680 (2017). https://doi.org/10.1007/s10846-016-0461-x
9. Kirkpatrick, S., Gelatt, C., Vecchi, M.: Optimization by simulated annealing. Science (New York, N.Y.) **220**, 671–80 (1983). https://doi.org/10.1126/science.220.4598.671
10. Liu, S., Li, X., Meng, M., Gong, X.: Energy-aware coverage path planning of multi-UAV based on relative distance scaling cluster method. In: 2023 International Conference on Advanced Robotics and Mechatronics (ICARM), pp. 709–714 (2023). https://doi.org/10.1109/ICARM58088.2023.10218969
11. Modares, J., Ghanei, F., Mastronarde, N., Dantu, K.: UB-ANC planner: energy efficient coverage path planning with multiple drones. In: 2017 IEEE International Conference on Robotics and Automation (ICRA), pp. 6182–6189, May 2017. https://doi.org/10.1109/ICRA.2017.7989732
12. Pitre, R.R., Li, X.R., Delbalzo, R.: UAV route planning for joint search and track missions-an information-value approach. IEEE Trans. Aerosp. Electron. Syst. **48**(3), 2551–2565 (2012). https://doi.org/10.1109/TAES.2012.6237608
13. Ren, J., Liu, K., Cui, Y., Du, W.: Search path planning algorithm based on the probability of containment model. Math. Probl. Eng. (2021). https://api.semanticscholar.org/CorpusID:234058033
14. San Juan, V., Santos Peñas, M., Andujar Marquez, J.: Intelligent UAV map generation and discrete path planning for search and rescue operations. Complexity **2018**, 1–17 (2018). https://doi.org/10.1155/2018/6879419
15. Tekever: ARX project. https://www.tekever.com/projects/arx/. Accessed 20 May 2024